сформувати певні напрями розвитку підприємств галузі як важливого чинника інтеграції нашої держави у світовий економічний простір.

На нашу думку, формування потужного стратегічного потенціалу розвитку підприємств водного транспорту та їх високого рівня конкурентоспроможності на ринку транспортних послуг можливе через забезпечення активної стратегічної взаємодії господарюючих суб'єктів.

Список використаних джерел:

1. Статистичний збірник «Транспорт та зв'язок України (2018)»: Державна служба статистики України, 2019. – 152 с.

2. Тимченко Н.М. Дослідження розвитку ринку перевезень водним транспортом в сучасних умовах господарювання / Н.М. Тимченко, О.Ю. Кузьменко. Науковий вісник Херсонської державної морської академії : науковий журнал. – Херсон : Херсонська державна морська академія, 2018. – № 2 (19). – С. 80–85.

Khrystova A.R.
Student,
Kharkiv National University of Radioelectronics

PARALLEL AND MULTITHREADED PROGRAMMING

Most programs are written for sequential execution. However, to achieve higher performance, multiprocessor computing systems are being created.

Unlike sequential computing programming, the conceptual basis of which is the concept of an algorithm implemented in steps strictly sequentially in time, in parallel programming, the program generates a set of parallel processes of information processing that are completely independent or related in a static or dynamic space or time -consequent relations.

Computational concurrency acts in various specific forms, depending on the programming stage, the complexity of the parallel fragments and the nature of the relationships between them.

In texts describing tasks and programs, it is possible to distinguish the levels of complexity of fragments for which the task of parallelization, ie. expressions over structural data (vectors, matrices, trees, etc.) that are written in algorithmic languages by loop operators; subtasks and subroutines; independent tasks and programs in multiprocessor systems.

The processes are parallel if they are executed simultaneously. Parallel processes can work completely independently of one another, or they can

periodically synchronize and interact. The main difficulty in designing parallel programs is to ensure the correct sequence of interactions between different computational processes, as well as to coordinate the resources shared between the processes.

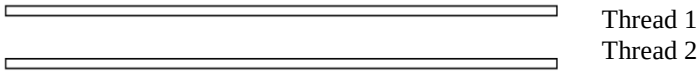

Figure 1. Parallel execution of the program

Source: [3]

Parallel programming is used to create programs that efficiently use computing resources by executing code on multiple computing nodes simultaneously.

Imagine the following picture: several cars go from Kiev to Lviv. Cars can fight for road space or go in the convoy, or overtake each other (getting into an accident). They can also drive on parallel lanes of the road and arrive almost simultaneously without crossing the road. Possible when all the cars go different routes and different roads. This picture demonstrates the essence of parallel computing: there are several tasks that need to be accomplished (machine driven). You can run them one by one on a single processor (road), in parallel on multiple processors (lanes) or on distributed processors (separate roads). However, tasks need to be synchronized to avoid collisions or delays at stop signs and traffic lights.

In addition to parallel programming, there is also a multi-threaded.

The thread specifically refers to the simultaneous execution of more than one series of instructions.

Thread is the programming of several parallel threads of execution. These threads can run on a single processor. Or there may be multiple threads running on multiple processor cores. The thread allows one processor to create multiple parallel threads. Each thread executes its own sequence of instructions. They all have access to the same shared memory space and communicate with each other as needed. Flows can be carefully managed to optimize performance.

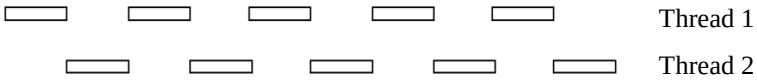

Figure 2. Multithreaded program execution

Source: [3]

This implementation of the program is beneficial when there are several tasks that can (at least partially) work simultaneously. The code of a properly written multi-threaded application looks simple because each thread performs its specific task.

On the other hand, multi-threaded applications are harder to write and debug. You will need to synchronize a lot of access to shared resources to avoid unpredictable results, as well as coordinate the execution of interdependent code to ensure the correct sequence of events.

And one last note: on a single-processor computer, multithreaded applications run faster than single-threaded ones. Speed only increases on a multiprocessor computer with a multiprocessor operating system.

So what's the difference between multithreading and concurrent programming? In the 21st century, most computers have multiple processor cores. This means that we can perform different tasks with different processors independently. Some problems are easy to make parallel, while others are completely consistent. For example, numerical methods are completely consistent: all steps depend on the previous steps.

So this is the main difference between multithreading and concurrent computing. No matter how many threads we use, if we only have one processor, then we need to use time quantization to service all threads. On the other hand, with multiple kernels, we can perform calculations simultaneously.

References:

1. Gorodnya L.V. Computing networks to describe the base language. – Sat. Parallel computing and software systems. – Novosibirsk, 1981. – p. 135-151.

2. N. Shavit, M. Herlihy. The Art of Multiprocessor Programming URL: http://molodyvcheny.in.ua/files/conf/other/41oct2019/41oct2019.pdf

3. Multithreading and Parallel Programming. Веб-сайт. URL: https://www.globalsoftwaresupport.com/multithreading-and-parallel-programming/